

Csoport neve: The Racoons

Feladat sorszáma: H_4

Feladat címe: Analízis Modell

Clean Paws Library elektronikus könyvtár

Gyakorlatvezető:

Árvai László

Csoport tagok:

Árvai László	TANKÖR	laszlo.arvai@uni-miskolc.hu
Kovács Dániel	F9Y7TW	kovacsd435@gmail.com
Mészáros István	WZZTLL	mesziist@gmail.com
Mester Dániel	FZIE3D	danielmester22@gmail.com
Fónagy Zeus Vilmos	G6HV2W	fonagyzeusz02@gmail.com
Kiss Tamás László	WN6YVX	tamaskiss2003@gmail.com

BEADÁS DÁTUMA

Történet

Dátum	Verzió	Leírás	Szerző
2023.11.21	0.1	Feladat Elkezdése	Mindenki
2023.11.22	0.2	Feladatok korrigálása	Mindenki
2023.11.16	1.0	Nyers domumentum elkészítése	Mindenki

1. Bevezetés

Az internet terjedésének köszönhetően bárki bármilyen információt szerezhet a világ bármely pontjáról. Régebbi időkben, az információk többségét könyvtárakból tudhatták meg az emberek ezért cégünk egy olyan szoftvert kínál fel a könyvtáraknak, amely segítségével búcsút mondhatnak a papíralapú vagy digitális táblázatokban tárolt leltáraiknak.

A Clean Paws Library segítik a könyvtárosokat és könyvtárakat a könyvek és egyéb könyvtári anyagok hatékony kezelésében és nyilvántartásában. A programunk célja az, hogy segítse a könyvtárakat a könyvek katalogizálásában, a kölcsönzések és visszavétel nyomon követésében, a könyvtárhasználók nyilvántartásában, és egyéb adminisztratív feladatokban.

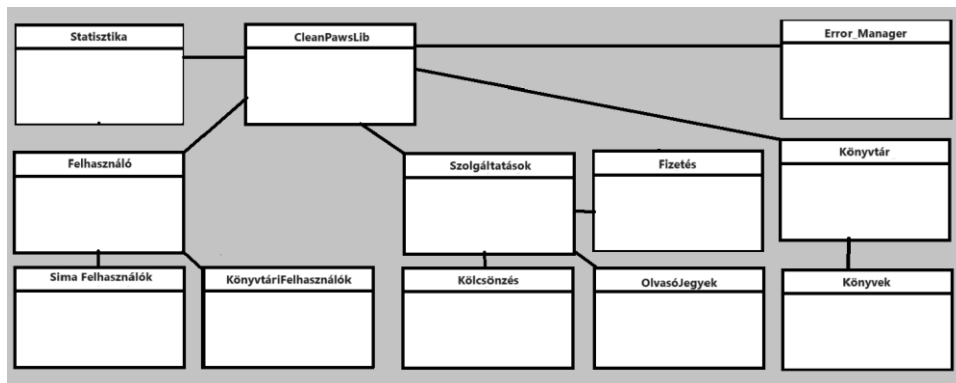
A program lehetővé teszi a könyvek és egyéb könyvtári anyagok részletes adatainak rögzítését, mint például cím, szerző, kiadó, ISBN, besorolás stb.

Az olvasók könnyen kereshetnek a könyvtár katalógusában, és kiválaszthatják a kölcsönzésre szánt könyveket és a program nyomon követi a kölcsönzéseket, a határidőket és a visszavételeket, meggyorsítva az ügyintézés. A végfelhasználók megújíthatják olvasójegyüket, valamint rendezhetik a késedelmi díjakat online is.

Az olvasni vágyókat egy egyszerű könnyen kezelhető webes felület fogadja, így a velünk szerződött könyvtáraknál lehetőség nyílik könyvek kölcsönzésére online felületen, valamint házhozszállítással segítünk olvasóinknak kedvenc könyvük elolvasásához.

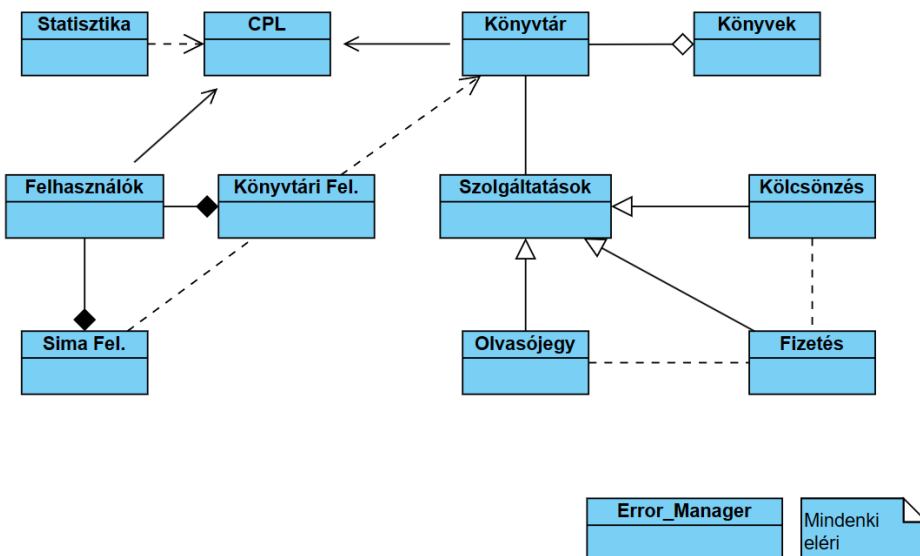
A könyvtárak részére biztosítjuk a tároláshoz szükséges szervereket ezért egyéb költség nem terheli őket. A szervereinken diszkréten kezeljük a könyvtár, illetve az olvasók adatait így gondoskodva a könyvtári adatok és felhasználói adatok védelméről.

2. Kezdeti osztálydiagram



Mint ahogy itt is látjuk, igazából nem kell sok osztály a program működtetéséhez. Kell pár a felhasználók tárolásának, illetve a fizethető dolgok tárolásának. A kihívás inkább a programon belül lévő metódusokkal történő adatbázis manipuláció foglya jelenteni.

2.1 Osztálydiagram



2.2 Osztályok felsorolása

Felhasználó:

Itt történik a felhasználók azonosítása és a rendszerhez való csatlakozásuk.

Sima Felhasználók:

A hétköznapi felhasználók itt tudják a számukra engedélyezett metódusokkal használni a rendszert.

Könyvtári Felhasználó:

Itt a rendszerben tárolt könyvtárak alkalmazottai tudnak hozzáférni a hozzájuk rendelt könyvtárakhoz.

CPL

A program fő osztálya. Összekapcsolja, vezérli a többi osztályt, melyek rajta keresztül kommunikálhatnak egymással.

Kölcsönzés:

A könyvek kölcsönzéséhez kapcsolódó folyamatok kezelése.

Könyv:

A rendszerben tárolt könyvek adatainak kezelése történik itt

Olvasójegyek:

Olvasójegyek adatainak és hozzájuk kapcsolódó folyamatok kezelése.

Könyvtár:

A rendszerben lévő könyvtárak adatainak és azok könyv készletének kezelése.

Szolgáltatások:

A felhasználók számára elérhető szolgáltatások "kölsönzés & olvasójegy" osztályainak vezérlése

Statisztika:

Adatbázis lekérdezése és a kapott adatból statisztikák elő állítása.

Fizetés:

Az online tranzakciók biztonságos kezelése és dokumentálása.

Error Manager:

A futás során fellépő, felhasználóktól érkező hibás adatok, hibák megfogalmazására szolgál.

2.3. Alrendszerek

Az elkészült osztálydiagram alapján a teljes rendszert funkcionálisan összetartozó osztályokból álló (funkcionális) alrendszerekre bontjuk. Ez a feladatok esetén általában néhány alrendszer. Tipikusan egy-egy alrendszer a felelős a megjelenítésért, a tényleges funkciók ("üzleti logika") végrehajtásáért, illetve az adatkezelésért. A feladat jellegéből adódóan a fenti alrendszereken kívül további alrendszerek is lehetnek. Az eredménye egy UML csomagdiagram, amely az egyes alrendszerek közötti együttműködést, információ áramlást is jelöli. Mintaként tekinthetjük az UML fóliasorozat 71. oldalán található példát.

Ezután egy-egy alpontban, amelyek címe "Alrendszernév alrendszer" (például "Vizsgáztatás alrendszer") formájú legyen, tartalmazza az egyes alrendszerek rövid leírását, és az alrendszerhez tartozó osztályok neveinek felsorolását.

Felhasználói felület alrendszer: Ez az alrendszer tartalmazza az összes olyan grafikus elemet, amely a szoftver a felhasználók felé való megjelenését szolgálja osztály diagrammon nem tűnik fel, mivel ez egy weboldal, így a grafikus megjelenítés nem része az osztály hierarchiának.

Adatbázis kezelő alrendszer: Az adatbázissal való kommunikálás, adatok tárolása és kezelésért felelős alrendszer. Itt mennek végbe a keresések az adatbázisokon. (Könyvek, Könyvtárak)

Szolgáltatásokat kezelő alrendszer: Ez kezeli a pénzel kiváltható szolgáltatásokat, mint például kölcsönzés, olvasójegy vétele, könyv idejének a hosszabbítása.

3. Felhasználói alrendszer modellje

3.1 Statikus modell

3.1.1 Kapcsolatok pontosítása

A felhasználók osztály feladata tárolni a felhasználók adatait, tartalmazza a felhasználókkal kapcsolatos metódusokat, alosztálya a könyvtári és a normál felhasználó. A felhasználók osztály kapcsolatban áll a könyvtár osztállyal, egy könyvtárhoz több felhasználó rendelhető.

3.1.2 Attribútumok azonosítása

Minden osztályhoz (vagy kapcsolathoz, ha ahhoz tartozik) hozzárendeljük a legfontosabb attribútumait. Ez az elemzés is vezethet új osztályok felfedezéséhez. (Például az OMT esettanulmány című segédletben a tornász és az egyesület kapcsolatának elemzése.)

Vázoljuk azokat a megfontolásokat, amelyek az osztálydiagram pontosításához vezetnek, és leírjuk, hogy mit változtat az eddigi információkon.

A felhasználók minden osztályának a specifikációjától függően tartalmaznia kell a szükséges paramétereket leíró attribútumokat. Ide tartozik például a könyvtár és a felhasználó esetében a cím, lakcím a `getDistance()` meghatározásához.

3.1.3 Bázisosztályok keresése

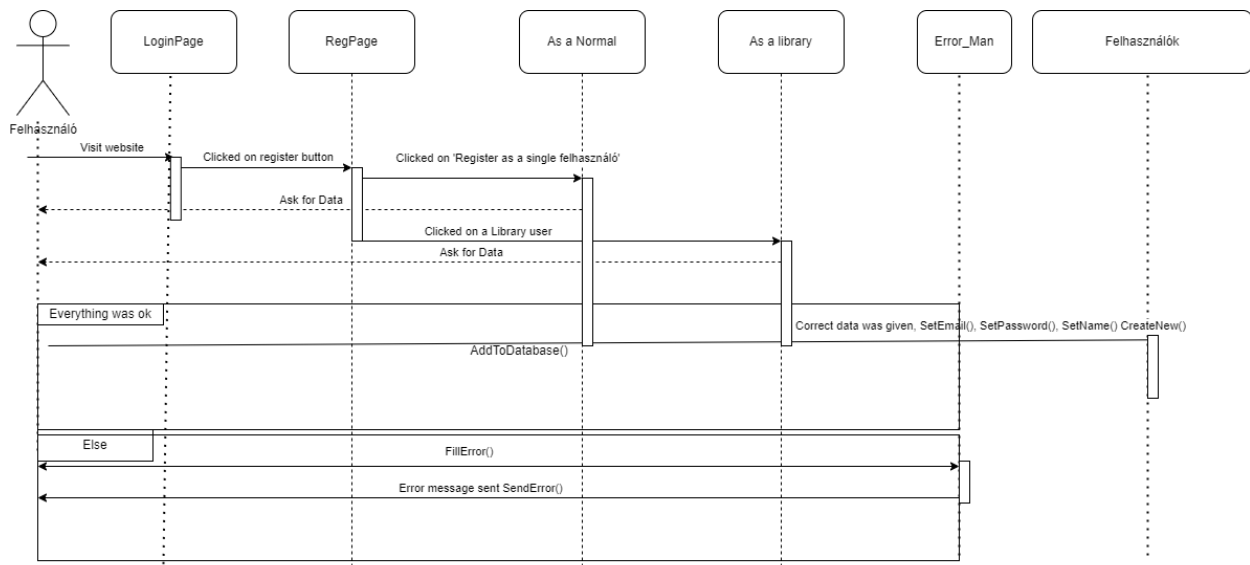
A könyvtári felhasználó és sima felhasználó osztályok közös argumentumait kiemelve, létrehozuk a `baseFelhasználó` bázis osztályt

3.2. Dinamikus modell

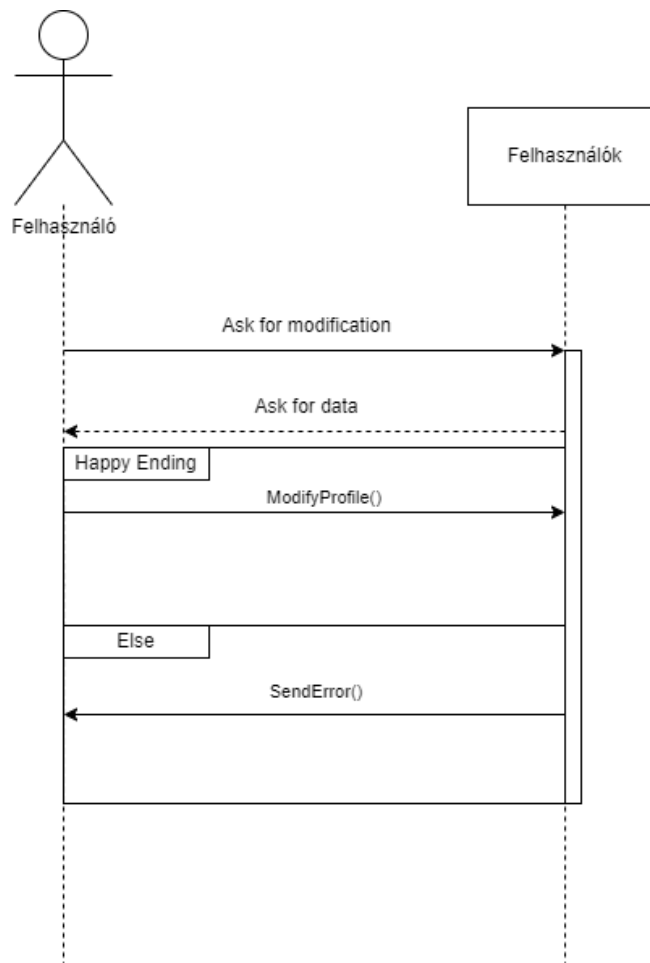
Egy-egy alpontban az SRS-ből átvett használati esetek szokásos működési eseteire elkészítjük a szükséges diagrammokat. A használati esetre a nevével hivatkozunk.

Ebben az alpontokban dokumentáljuk a szükséges állapotdiagramokat is. Az állapotdiagramok ábra aláírásában szerepeljen a vonatkozó osztály neve.

Ha az áttekinthetőséget javítja, az alpont további alpontokra tagolható.



Bejelentkezés/regisztráció szekvenciadiagrammja



Felhasználó módosításának a szekvenciadiagrammja

3.3. Funkcionális modell

Egy – egy alponthan dokumentáljuk leírással vagy a szükséges diagrammokkal a rendszerben zajló legfontosabb feldolgozási lépéseket, az egyes tevékenységek között szükséges adat áramlásokat.

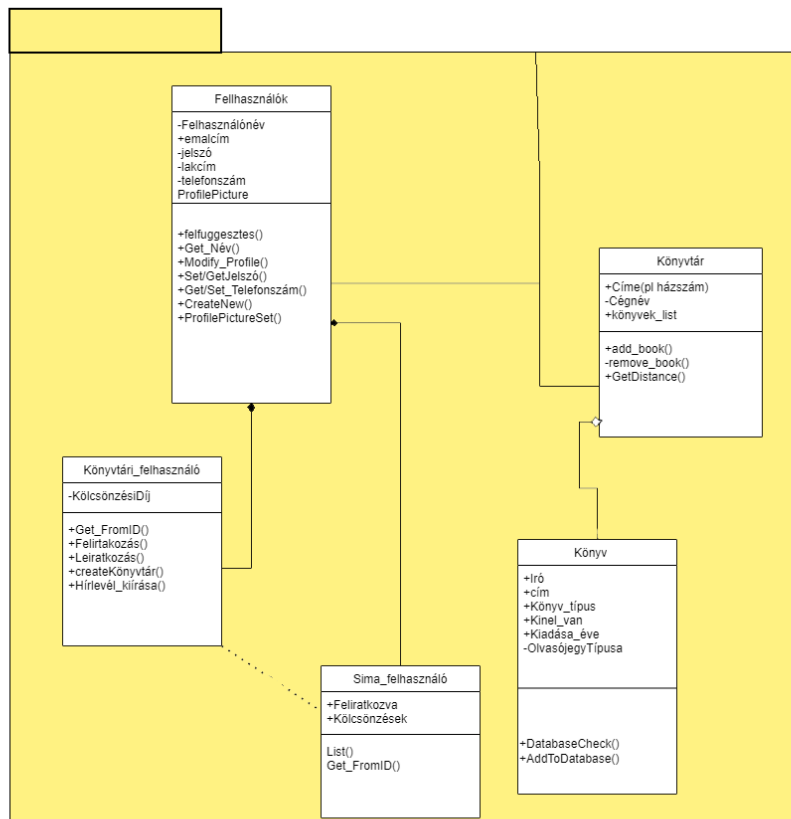
Ha az áttekinthetőséget javítja, az alpont további alpontokra tagolható.

Ez az alrendszer a felhasználókat kapcsolja össze a megfelelő könyvtárral, a segítségével tud a felhasználó és a könyvtár adatokat egyeztetni, elérni egymás adatait, metódusait, funkcióit.

3.4. Operációk azonosítása

Mivel az eltárolt paraméterek döntő többsége nem vehet fel akármilyen értéket, módosításuk csakis metódusokon keresztül valósulhat meg. Ezek a set szóval kezdődő metódusok. Ezek ellenőrzik az érték helyességét. Az adatelrejtés elve miatt ezeket a paramétereket privát adattagokban tároljuk el, így közvetlen módon más osztályból nem elérhetőek. A hozzáférés get szóval kezdődő metódusokon keresztül valósul meg.

3.5 Az analízis modell osztálydiagramja



3.6. Az analízis modell osztályainak listája

Az előző pontban szereplő osztálydiagram osztályainak leírása. Minden osztály leírását kiegészítjük az attribútumokkal és az operációkkal. Így egy alpont az alábbi formájú lesz:

Felhasználók: A felhasználó anyaosztályának, vagy más néven a gyökér osztálynak, kiemelkedő szerepe van egy alkalmazásban, mivel ő tartalmazza a fontosabb adatokat, amelyek részletesen leírják a felhasználói entitást. Ezen osztály rendszerint magában foglalja azokat az alapvető attribútumokat és tulajdonságokat, amelyek alapján azonosítani és kezelni lehet a felhasználókat.

Tulajdonságai:

<i>Név</i>	Típus	Leírás
Felhasználónév	Szöveg	A felhasználók neve
Jelszó	Szöveg	A felhasználók bejelentkezéséhez szükséges jelszó
Emailcím	Szöveg	A felhasználó digitális levelezési címe
Lakcím	Szöveg	A felhasználó lakcíme
Telefonszám	Szöveg	A felhasználó telefonszáma
Profile Picture	Szöveg	A felhasználó profile-ját tartalmazza linkel megadva

Operációk / Metódusok:

<i>Név</i>	Argumentumok	Működése, feladata
Felfüggesztés ()	Felhasználó ID	Felhasználók felfüggesztése
Get_Név()		Visszaadja a felhasználó nevét
Modify_Profile()	Stringek, melyek a változtatott adatok lesznek	A felhasználói adatok módosítása
Set/Get_Jelszó()		Jelszóváltoztatás

Könyvtári felhasználó: A könyvtárként regisztrált felhasználók objektuma az alkalmazás fontosabb entitásainak egyike, és szoros kapcsolatban áll a könyvtár és a könyv osztályokkal. Az ilyen felhasználókat az alkalmazásban való regisztráció során könyvtárkezelőként azonosítjuk, és számukra lehetőség nyílik könyvtári funkciókhoz és könyvek kezeléséhez.

Tulajdonságai:

<i>Név</i>	Típus	Leírás
KölcsönzésiDíj	int	A havi kölcsönzés értéke

Operációk / Metódusok:

<i>Név</i>	Argumentumok	Működése, feladata
GetFromID()	int	Kiírás ID-ből
Feliratkozás()	FelhasználóID	Lehetővé teszi, hogy az igénylő lássa a hírlevelet
Leiratkozás()	FelhasználóID	Lehetővé teszi, hogy az igénylő ne lássa a hírlevelet
CreateKönyvtár()	Stringek, melyeket a regisztrációnál kell megadni	Könyvtárat csinál a felhasználóhoz
HírlevélKiírása()	String, illetve egy link	Kiírat egy hírlevelet.

3.6.1 CPL

- **felelőssége, feladata:** A program fő osztálya. Összekapcsolja, vezérli a többi osztályt, melyek rajta keresztül kommunikálhatnak egymással.
- **együtműködők:** Könyvtár, Szolgáltatások, Felhasználók

Attribútumok

<i>Név</i>	Típus	Leírás
ID	Integer	A könyvtárak egyedi azonosító kódja.
KönyvtárID	Integer	A könyvtárak azonosítására használt változó
FelhasználóID	Integer	A felhasználók azonosítására használt változó

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
List()		Kilistáz
Sort()		Sorbarendez
AddToDatabase()		Hozzáadás az adatbázishoz
RemoveFromDatabase()		Eltávolítás az adatbázisból

3.6.2. Felhasználók

- **felelőssége, feladata:** Itt történik a felhasználók azonosítása és a rendszerhez való csatlakozásuk.
- **együtműködők:** CPL, Könyvtári_felhasználók, Sima_felhasználók

Attribútumok

<i>Név</i>	Típus	Leírás
Felhasználónév	Szöveg	A felhasználók neve
Jelszó	Szöveg	A felhasználók bejelentkezéséhez szükséges jelszó
Emailcím	Szöveg	A felhasználó digitális levelezési címe
Lakcím	Szöveg	A felhasználó lakcíme
Telefonszám	Szöveg	A felhasználó telefonszáma

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
Felfüggesztés ()	Felhasználó ID	Felhasználók felfüggesztése
Get_Név()		Visszaadja a felhasználó nevét
Modify_Profile()		A felhasználói adatok módosítása

Set/Get_Jelszó()		Jelszóváltoztatás
Get/Set_Telefonszám()		Telefonszám lekérése módosítása

3.6.3. Könyvtári_Felhasználó

- **felelőssége, feladata:** Itt a rendszerben tárolt könyvtárak alkalmazottai tudnak hozzáférni a hozzájuk rendelt könyvtárakhoz.
- **együttműködők:** CPL, Felhasználók,

Attribútumok

<i>Név</i>	Típus	Leírás
Kölcsönzési díj	Integer	A könyvek kölcsönzési díja

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
GetFromID()		FelhasználóID lekérése
Feliratkozás()		Felhasználók fel iratkoztatása a hírlevélről
Leiratkozás()		Felhasználók le iratkoztatása a hírlevélről
CreateKönyv()		Könyvek hozzáadása
Hírlevél_kiadása()		Hírlevél küldés a felhasználóknak

3.6.4 Sima_Felhasználó

- **felelőssége, feladata:** A hétköznapi felhasználók itt tudják a számukra engedélyezett metódusokkal használni a rendszert.
- **együttműködők:** CPL, Felhasználók,

Attribútumok

<i>Név</i>	Típus	Leírás
Fel iratkozva	Logikai	A hírlevél felíratkozását jelzi
Kölcsönzések	Szöveg	Kikölcsönzött könyvek

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
List()		Kilistáz

GetFromID ()		FelhasználóID lekérése
--------------	--	------------------------

3.6.5 Könyvtár

- **felelőssége, feladata:** A rendszerben lévő könyvtárak adatainak és azok könyv készletének kezelése.
- **együtműködők:** CPL, Szolgáltatások, Könyvek

Attribútumok

<i>Név</i>	Típus	Leírás
Cím	Szöveg	A könyvtár címe
Cégnév	Szöveg	A könyvtár neve
Könyvek_list	Szöveg	

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
AddBook()		Könyv hozzáadása az adatbázishoz
Remove_book()		Könyv eltávolítása az adatbázisból
GetDistance()		A felhasználó és a könyvtár közötti távolság

3.6.6 Könyvek

- **felelőssége, feladata:** A rendszerben tárolt könyvek adatainak kezelése történik itt.
- **együtműködők:** CPL, Szolgáltatások, Könyvtár

Attribútumok

<i>Név</i>	Típus	Leírás
Író	Szöveg	A könyv szerzője
Cím	Szöveg	A könyv címe
Könyv_típus	Szöveg	A könyv kategóriája
Kinél_van	Szöveg	Melyik felhasználó kölcsönözi a könyvet
Kiadás_éve	Dátum	Megjelenés éve
OlvasójegyTípusa	Szöveg	Olvasójegy típusa

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
DatabaseCheck()		Adatbázis ellenőrzése a könyv meglétéről

AddToDatabase()		Hozzáadás adatbázishoz
-----------------	--	------------------------

4. Szolgáltatások alrendszer modellje

4.1.1 Kapcsolatok pontosítása

A szolgáltatások osztály feladata az olvasójegyeket összekapcsolni a kölcsönzésekkel és az ehhez szükséges fizetési lehetőségeket a könyvtárakhoz. A könyvtárakhoz tartozó olvasójegyek összekapcsolja felhasználókkal, valamint figyeli a kölcsönzési idő lejáratát dátumát. Lehetőséget kínál meghosszabbítani a kölcsönzést.

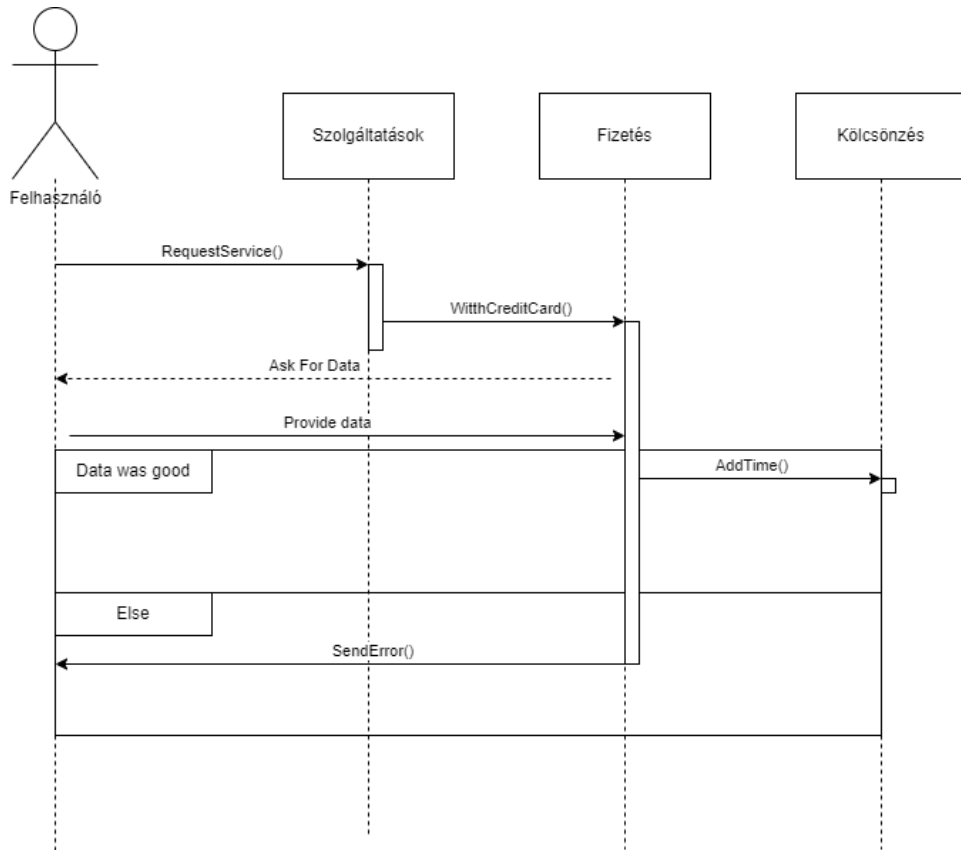
4.1.2 Attribútumok azonosítása

A szolgáltatások alrendszer minden osztályának a specifikációjától függően tartalmaznia kell a szükséges paramétereket leíró attribútumokat. A felhasználók alapján azonosíthatók az általa használt, igényelt szolgáltatások pl. Van-e olvasójegye, kölcsönzött-e könyvet.

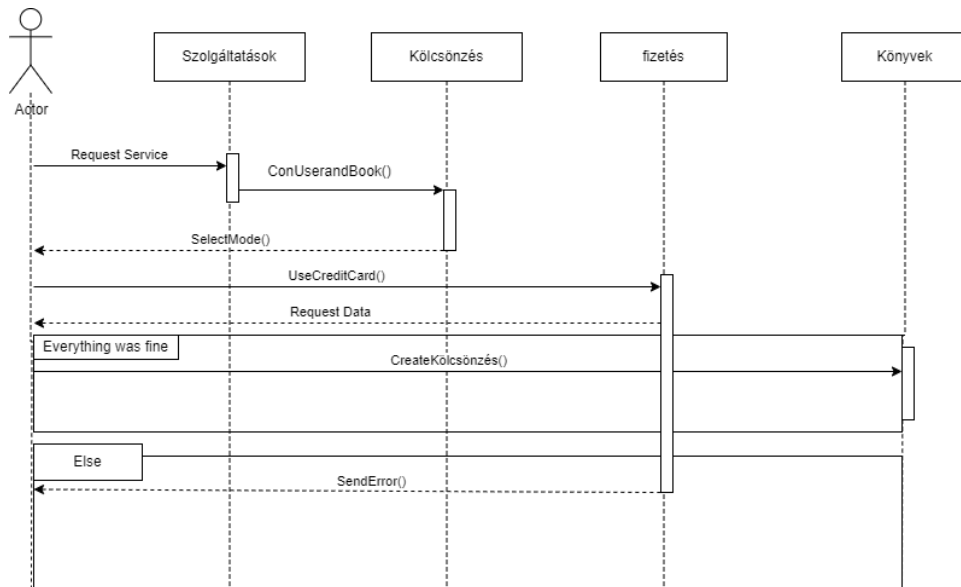
4.1.3 Bázisosztályok keresése

A szolgáltatások alatt lévő osztályok hasonló argumentumokkal rendelkeznek, célszerű alosztályokként kezelni őket az átláthatóság és a megfelelő adatintegritás miatt. Viszont kiemelhetők egy nagyobb osztályként ez lenne a Szolgáltatások, amelyből származtatjuk a többi alosztályt.

4.2. Dinamikus modell



Kölcsönzési idő módosítása



Könyvek kölcsönzése

4.3. Funkcionális modell

Egy – egy alponban dokumentáljuk leírással vagy a szükséges diagrammokkal a rendszerben zajló legfontosabb feldolgozási lépéseket, az egyes tevékenységek között szükséges adat áramlásokat.

Ha az áttekinthetőséget javítja, az alpont további alpontokra tagolható.

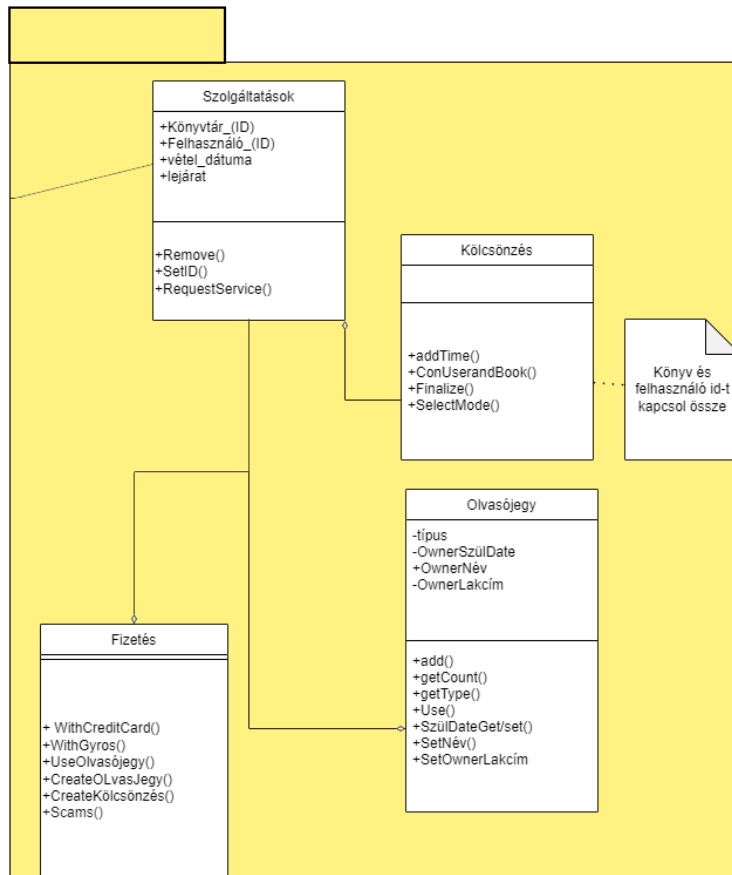
4.4. Operációk azonosítása

A dinamikus és funkcionális modell felhasználásával az osztálydiagramot kiegészítjük az egyes osztályokhoz tartozó operációkkal. Eközben újabb osztályok, kapcsolatok és attribútumok szükségességére is fény derülhet.

Ha szükséges, dokumentáljuk azokat a meggondolásokat és döntéseket, amelyek az osztálydiagram kialakításához vezettek.

Ha az áttekinthetőséget javítja, az alpont további alpontokra tagolható.

4.5 Az analízis modell osztálydiagramja



4.6 Az analízis modell osztályainak listája

4.6.1 Szolgáltatások

- felelőssége, feladata:** A felhasználók számára elérhető szolgáltatások “kölcsönzés & olvasójegy” osztályainak vezérlése
- együtműködők:** CPL, Könyvtár, Kölcsönzés, Olvasójegy, Fizetés

Attribútumok

Név	Típus	Leírás
KönyvtárID	Integer	A könyvtárak egyedi azonosító kódja.
FelhasználóID	Integer	A felhasználók egyedi azonosító kódja.
Vétel_Dátuma	Dátum	Kezdeti dátum
Lejárat	Dátum	Lejárat dátum

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
Remove()		Eltávolítás
SetID()		ID beállítása

4.6.2 Kölcsönzés

- **felelőssége, feladata:** A könyvek kölcsönzéséhez kapcsolódó folyamatok kezelése.
- **együtműködők:** CPL, Könyvtár, Szolgáltatások

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
AddTime()		Időtartam beállítása
ConUserandBook()		Felhasználó és könyv összekapcsolása

4.6.3 Olvasójegy

- **felelőssége, feladata:** Olvasójegyek adatainak és hozzájuk kapcsolódó folyamatok kezelése.
- **együttműködők:** CPL, Könyvtár, Szolgáltatások

Attribútumok

<i>Név</i>	Típus	Leírás
Típus	Szöveg	Az olvasójegy típusa
OwnerSzülDate	Dátum	A felhasználó születési dátuma
OwnerNév	Szöveg	A felhasználó neve
OwnerLakcím	Szöveg	A felhasználó lakcíme

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
Add()		Olvasójegy létrehozása
GetCount()		
GetType()		
Use()		
SzülDateGet/Set()		Születési dátum lekérése/beállítása
SetNév()		Név beállítása
SetOwnerLakcím()		Lakcím beállítása

4.6.4 Fizetés

- **felelőssége, feladata:** Az online tranzakciók biztonságos kezelése és dokumentálása.
- **együttműködők:** CPL, Könyvtár, Szolgáltatások

Attribútumok

<i>Név</i>	Típus	Leírás
Ár	Integer	Az összeg tárolása

•

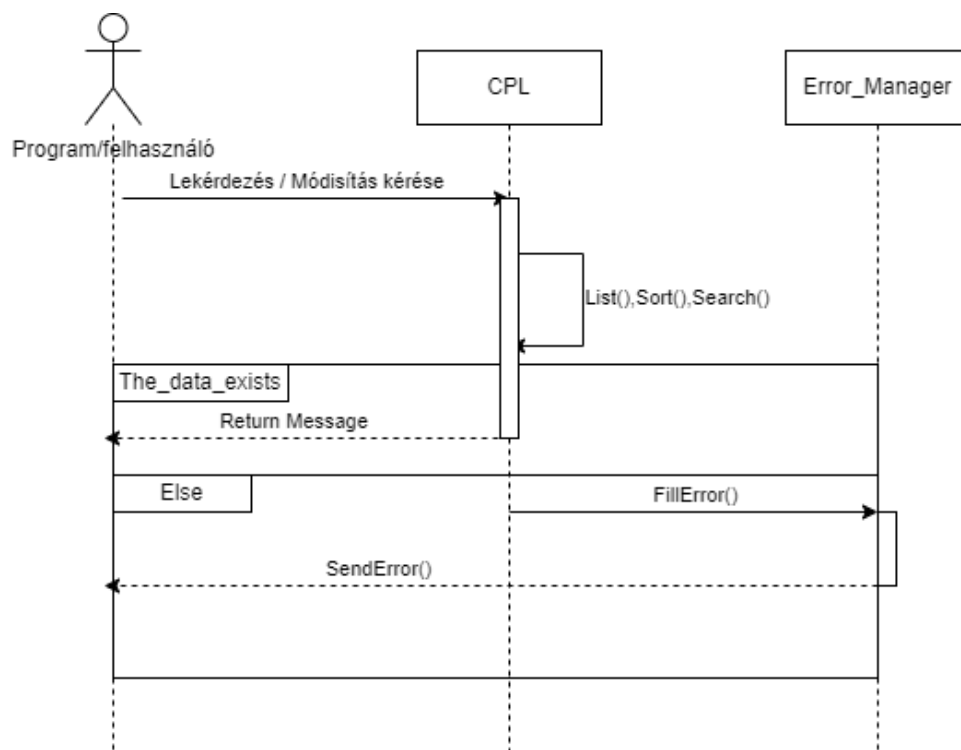
Operációk

<i>Név</i>	Argumentumok	Működése, feladata
WithCreditCard()		Fizetés bankkártyával
WithGyros()		Fizetés étellemmel

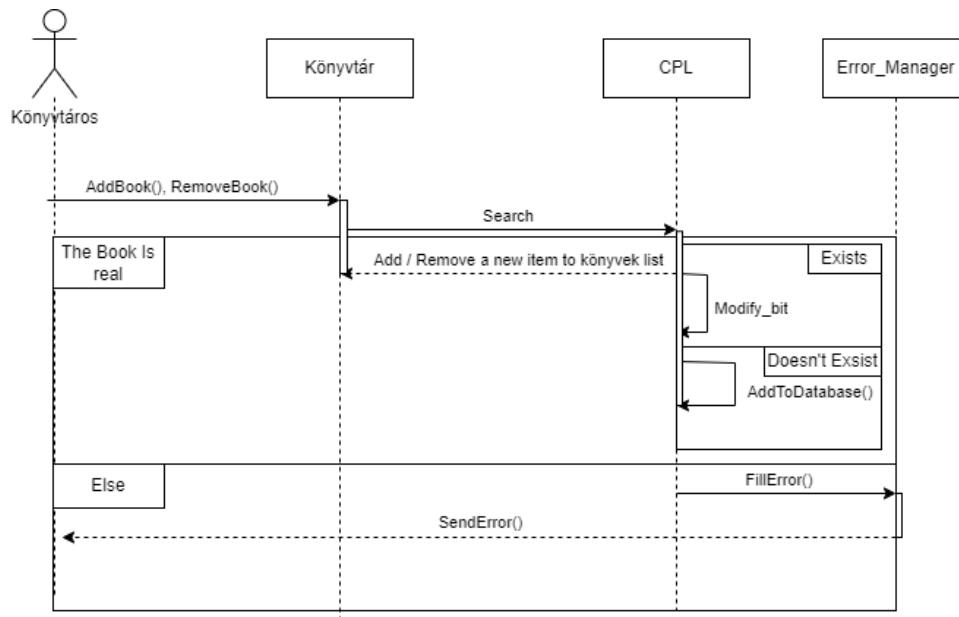
UseOlvasójegy()		Olvasójegy használata
CreateOlvasójegy()		Olvasójegy létrehozása
CreateKölcsönzés()		Kölcsönzés indítványozása
Scams()		A fejlesztők fizetés kiegészítése

5. Statisztikai alrendszer modellje

5.1 Dinamikus Modell



Lekérdezések, adatbázis módosítások



Új könyv hozzáadása/levétele az adatbázisból

5.1.1 Kapcsolatok pontosítása

A Statisztika osztálynak a könyvtárban található adatok alapján kell szolgáltatnia statisztikai információkat, az ID alapján megkülönböztetjük a különböző könyvtárakat, a CPL főosztályon keresztül érhetünk el adatokat a könyvtárakról.

5.1.2 Attribútumok azonosítása

Az adatbázis lekérdezéseket felhasználva, a statisztikai metódusokat tárolja, lekérdezhetőek az adminok által.

5.2. Dinamikus modell

Egy-egy alponthoz az SRS-ből átvett használati esetek szokásos működési eseteire elkészítjük a szükséges diagrammokat. A használati esetre a nevével hivatkozunk.

Ebben az alponthoz dokumentáljuk a szükséges állapotdiagramokat is. Az állapotdiagramok ábra aláírásában szerepeljen a vonatkozó osztály neve.

Ha az áttekinthetőséget javítja, az alpont további alpontokra tagolható.

5.3. Funkcionális modell

Egy – egy alponthban dokumentáljuk leírással vagy a szükséges diagrammokkal a rendszerben zajló legfontosabb feldolgozási lépéseket, az egyes tevékenységek között szükséges adat áramlásokat.

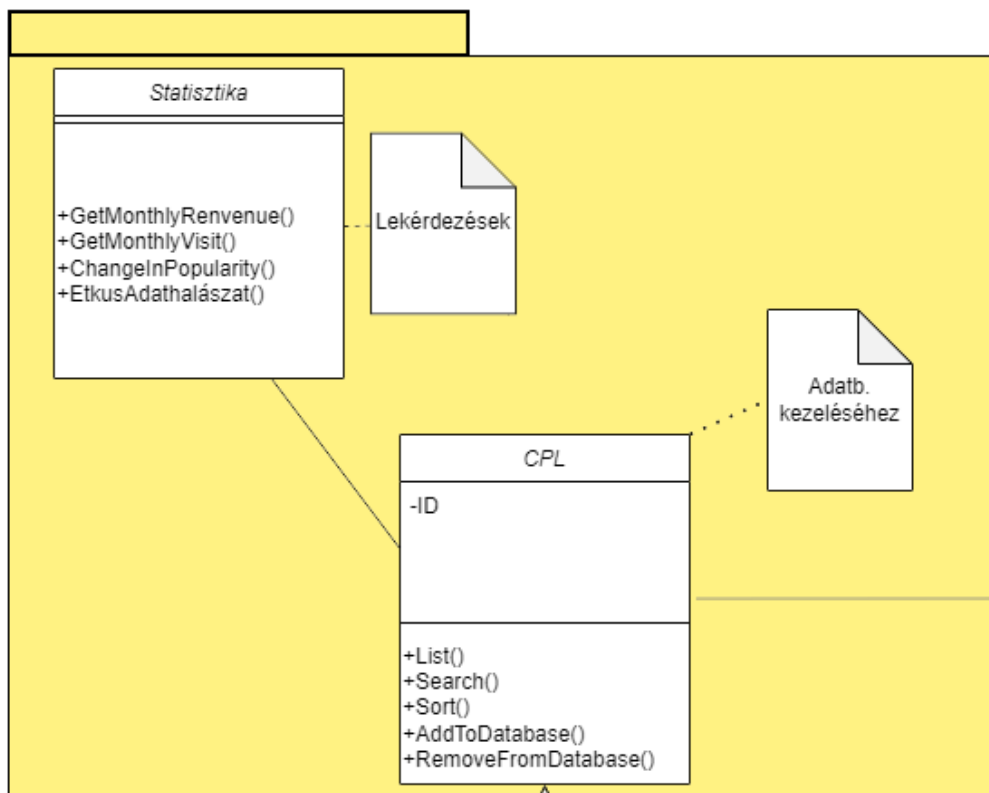
Ha az áttekinthetőséget javítja, az alpont további alpontokra tagolható.

5.4. Operációk azonosítása

A Statisztika osztály operációinak azonosítása során különböző statisztikai funkciókat és műveleteket azonosíthatunk, amelyeket a könyvtár adatbázisával végrehajt. Az alábbiakban néhány példa a lehetséges operációkra:

- GetMonthlyRevenue: Havi bevételt visszaadó függvény
- GetMonthlyVisit: Havi látogatószámot visszaadó függvény
- ChangeInPopularity: Könyv népszerűségét visszaadó függvény

5.5 Az analízis modell osztálydiagramja



5.6 Az analízis modell osztályainak listája

5.6.1 Statisztika

- **felelőssége, feladata:** Statisztikai adatokat tárol.
- **együtműködők:** CPL,
- **Attribútumok**

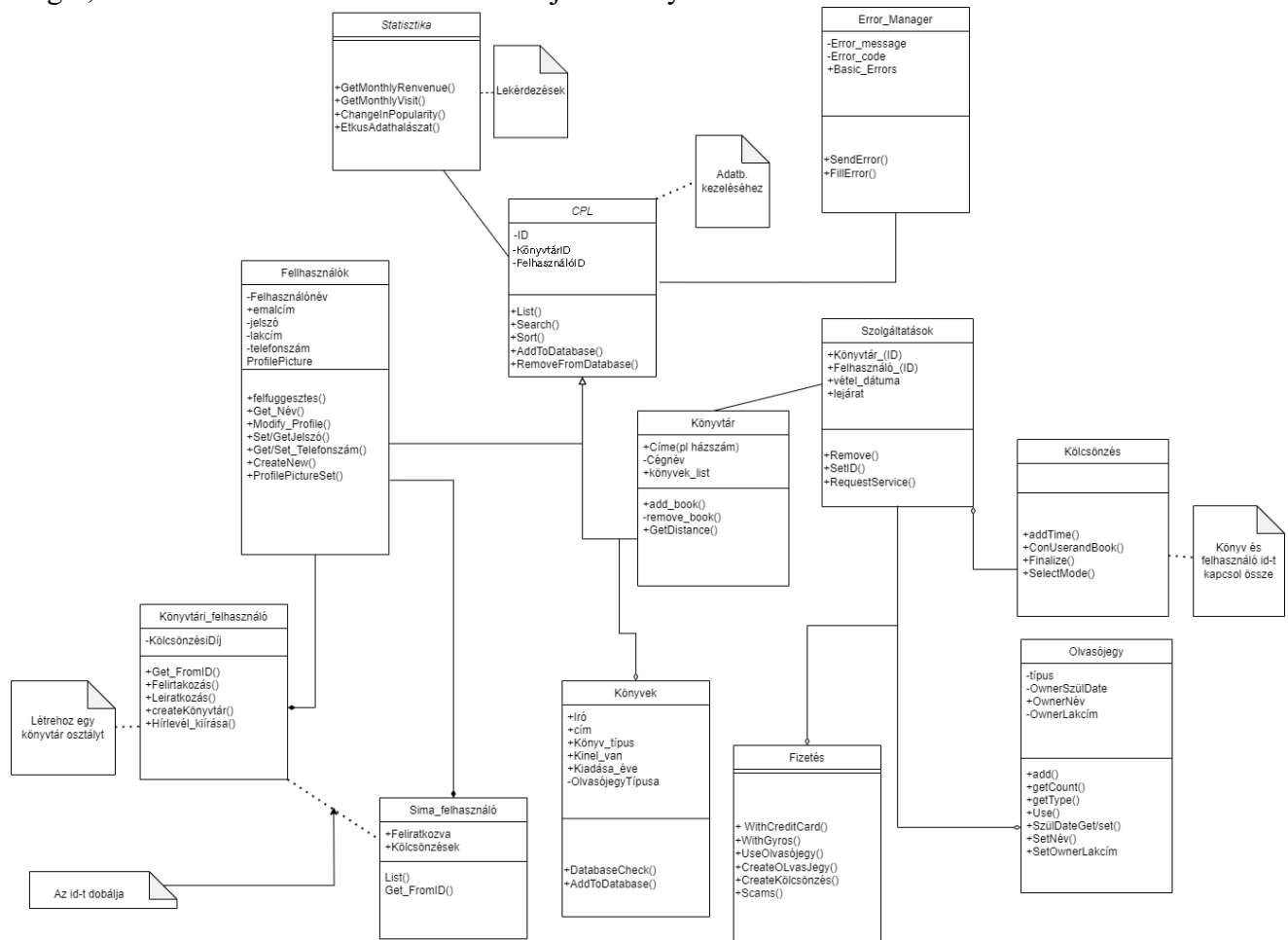
<i>Név</i>	Típus	Leírás
Bevétel	Integer	Tárolja a bevételt
Látogatók_száma	Integer	A látogatók számának tárolása

Operációk

<i>Név</i>	Argumentumok	Működése, feladata
GetMonthlyRevenue()		Havi bevételről ad visszajelzést
GetMonthlyVisit()		A havi látogatószámot mutatja meg
ChangeInPopularity()		
EtikusAdathalászat()		Adatok begyűjtése

6. A teljes, végleges osztálydiagramm

Végül, a fenti alrendszerekből összeáll a teljes osztályrendszer:



Itt a szemfüles ember felfedezhet egy utolsó osztályt, ami eddig nem szerepelt sehol, ami nem más, int az Error_Manager, ami a hibák kezeléséért felelős, ezért minden osztálynak el kell tudni érnie, ezért nem szerepel egy alrendszerben se.

Attribútumai:

- Error_Message: A hiba szövege
- Error_Code: Az alapból létrehozott hibák kódjai
- Basic_errors: Az alapból létrehozott hibák listája

Metódusok:

- SendError()-hiba küldése
- FillError()-Hiba megírása, ami küldve lesz